Zend_RepoMon Proposal

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_RepoMon Component Proposal

Proposed Component Name	Zend_RepoMon
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_RepoMon
Proposers	Eric David
Revision	1.1 - 27 December 2007: Initial draft. (wiki revision: 25)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_RepoMon is a tool that identifies and reports programatic modifications made to PHP and JavaScript files within a version control system.

2. References

- Subversion

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** report the addition, deletion or signature modification of a class, method or independent function within PHP and JavaScript files.
- This component **will** be able to generate reports in either plain text, bare (CSS-ready) XHTML 1.1 or XHTML with inline styles.
- The initial release **will** only support SVN, but will be built scalable to be able to add CVS and other versioning systems support.
- This component **will** be able to automatically email user-specific reports on a periodic basis when used in tandem with an instantiating script and crontab.
- This component **will** be able to accept revision boundaries as revision numbers, versioning system constants (head, prev, etc.) or any valid date/time format as defined by strtotime().
- This component **will not** in any way modify files within the versioning system. For extra security, users are encouraged to utilize a read-only versioning system user account with RepoMon.

4. Dependencies on Other Framework Components

- Zend_Exception

5. Theory of Operation

RepoMon abstracts data gathered from the versioning system and generates a report of modifications to classes, method signatures and functions. Reports can be displayed via the browser or from the command line and can be sent via email as well. Various reporting output formats can be selected. The XHTML 1.1 format allows complete control over the display appearance via CSS.

6. Milestones / Tasks

- Milestone 1: Finalize a working draft of this proposal suitable for a basis to guide development
- Milestone 2: Establish a functional module within the Zend framework based on the initial Component Requirements, Constraints, and Acceptance Criteria
- Milestone 3: Debug and create test cases
- Milestone 4: Add support for other output formats such as JSON and XML
- Milestone 5: Enhance, improve and expand functionality based on feedback from Zend community
- Milestone 6: Add support for other versioning systems

7. Class Index

- Zend_RepoMon
- Zend_RepoMon_Exception
- Zend_RepoMon_EmailSender (This could possibly be replaced with Zend_Mail)

8. Use Cases

All use cases need a username, password and base path to repository. For brevity the use cases assume the following precedes them (and is substituted with values from your versioning system.)

```
$repoMonConfig = array('username' => 'yourUsername',
                       'password' => 'yourPassword',
                       'basePathToRepository' => 'http://svn/svn');
```

UC-01

Get revisions since yesterday

```
$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath('manage/trunk');

echo $RepoMon->showRevisions('yesterday');
```

UC-02

Get all the revisions from the previous week

```
$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath('manage/trunk');

echo $RepoMon->showRevisions('last sunday - 1 week', 'last sunday');
```

UC-03

Get revisions made to the Zend framework in the previous week

```
$reportPath = 'manage/trunk/library/Zend';

$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath(isset($reportPath) ? $reportPath : 'manage/trunk');

echo $RepoMon->showRevisions('last sunday - 1 week', 'last sunday');
```

UC-04

Send a text-based email containing November revisions

```
$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath(isset($reportPath) ? $reportPath : 'manage/trunk');

$RepoMon->setReturnFormat('text');
$RepoMon->setEmail(array('recipient' => 'no-reply@zend.com'));

echo $RepoMon->showRevisions('Nov. 2006', 'Dec. 2006');
```

UC-05

Send an HTML-based email formatted for legacy mail clients that do not support CSS

```
$revisionDateBegin = 'last month';

$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath(isset($reportPath) ? $reportPath : 'manage/trunk');

$RepoMon->setReturnFormat('html');
$RepoMon->setEmail(array('recipient' => 'no-reply@zend.com',
                         'senderEmail' => 'no-reply@php.net'),
                         'senderName' => 'John Doe',
                         'subject' => 'Modification digest for the ' .
$revisionDateBegin);

echo $RepoMon->showRevisions($revisionDateBegin);
```

UC-06

Display nicely formatted XHTML 1.1 modifications

```
$RepoMon = new RepoMon($repoMonConfig['username'],  $repoMonConfig['password'],
$repoMonConfig['basePathToRepository']);
$RepoMon->setReportPath(isset($reportPath) ? $reportPath : 'manage/trunk');

$header = '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">'
            . '<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">'
            . '<head><title>RepoMon</title>'
            . '<link type="text/CSS" rel="stylesheet" href="example.css">';
            . '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/></head><body>';
$RepoMon->setHeader($header);
$RepoMon->setFooter('</body></html>');

echo $RepoMon->showRevisions('last week', 'head');
```

example.css

```
.repomon div {
    margin-bottom: 10px;
}
.repomon div span {
    font-style: italic;
}
.repomon .modified_element {
    margin-left: 20px;
}
```

9. Class Skeletons

```
class Zend_RepoMon_Exception extends Zend_Exception {}

class Zend_RepoMon {

    /**
     * Set the minimum required versioning system data
     *
     * @param string $username
     * @param string $password
     * @param string $basePathToRepository
     */
     public function __construct() {}

    /**
     * This overrides the default versioning system type of svn
     *
     * @param string $versioningSystemType
     */
     public function setVersioningSystemType($versioningSystemType) {}

    /**
     * Retrieve the contents of a file from the versioning system based on filename
and revision number
     *
     * @param string $filename
     * @param string $revisionNumber
     * @return string
     */
     public function getFileContentsByRevision($filename, $revisionNumber) {}

    /**
     * Get the class names that appear in a file along with their line numbers
     *
     * @param string $content
     * @return array
     */
     public function scrapeFileContentForClasses($content) {}

    /**
     * Extracts aggregate data from the raw-revisions string and populates
$this->_modifications
     *
     * @param array $rawRevisions The versioning system generated diff data
     */
     public function scrapeRevisions($rawRevisions) {}

    /**
     * Retrieve the revisions
     *
     * @param string $versioningSystemCommand Default is diff
     * @return array Each line of data as a seperate element in the array
     */
     public function getRevisions($versioningSystemCommand = 'diff') {}
```

```
    /**
     * Set the range of revisions to track
     *
     * Valid options are:
     * Any date and/or time string in virtually any format, all of the following are
valid:
     *     yesterday, last wednesday, -48 hours, Dec 1, 2006
     * Revision numbers (must be passed as an integer, not a string)
     * Any of the following string literals:
     *     head, base, commited, prev
     *
     * @param mixed $revisionBegin
     * @param mixed $revisionEnd optional property,
     *     if left out revisions will end at the current repository state
     * @return string
     */
     public function showRevisions($revisionBegin, $revisionEnd = null) {}

    /**
     * Attempt to prepend the class name before a method name
     * If not part of a class nothing will be changed.
     *
     * @param string $filename
     * @param int    $linenumber
     * @param string $action Must be either added or removed
     * @param mixed $objectName Either a string or an associative array of strings
     * @return string
     */
     protected function _prependClassName($filename, $lineNumber, $action,
$objectName) {}

    /**
     * Add a modification to the _modifications array
     *
     * @param string $filename
     * @param int    $linenumber
     * @param string $pluralizedString
     * @param string $action Must be either added or removed
     * @param mixed $objectName Either a string or an associative array of strings
     * @return void
     */
     protected function _addModification($filename, $lineNumber, $pluralizedString,
$action, $newObject) {}

    /**
     * Verifies if the format of the revision boundary is valid as-is;
     * if not it attempts to reformat it into a valid date string
     *
     * @param mixed $val
     * @return mixed
     * @throws Zend_RepoMon_Exception if the format cannot be parsed
     */
     protected function _formatRevisionBoundary($val) {}

    /**
     * Helper method to format the array of data into XHTML
     *
     * @return string Formatted results string
     */
```

```
     protected function _formatScrapedModificationsXhtml() {}

    /**
     * Helper method to format the array of data into HTML with inline formatting
     * This was created to accommodate legacy mail clients that do not support CSS
     *
     * @return string Formatted results string
     */
     protected function _formatScrapedModificationsHtml() {}

    /**
     * Helper method to format the array of data into humanly-readable text
     *
     * @return string Formatted results string
     */
     protected function _formatScrapedModificationsText() {}

     /**
     * Switchboard to appropriately format the array of data and set the default email
output format
     * Default return format is XHTML
     *
     * @return string Formatted results string
     */
     protected function _formatScrapedModifications() {}

    /**
     * Sets the revision boundaries
     *
     * @param mixed $revisionBegin
     * @param mixed $revisionEnd Optional property
     */
     protected function _setRevisionBoundaries($revisionBegin, $revisionEnd = null) {}

    /**
     * This handles all the setMyVar() methods that are not explicitly defined
     *
     * @param string $name The name of the attempted method call
     * @param array $value Arguments of the method being called
     */
```

```
        protected function __call($name, $value) {}
    }
```

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>